US-PAT-NO: 5455946

DOCUMENT-IDENTIFIER: US 5455946 A

TITLE: Method and means for archiving modifiable pages in a log based transaction management system

DATE-ISSUED: October 3, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mohan; Chandrasekaran	San Jose	CA	N/A	N/A
Narang; Inderpal S.	Saratoga	CA	N/A	N/A

US-CL-CURRENT: 707/202, 707/204 , 711/161 , 714/20

ABSTRACT:

A method and means for achieving **files** of modifiable pages in a log based phased commit transaction management system (TMS) in which those pages which have been modified since the last full or **incremental backup** donot require during the copy operation any modifications to the page itself but merely to a common status page. This is accomplished by management of a pair of global log sequence numbers. Comparison between a first number (ICBU.sub.-- LSN) and each data page LSN as the page is modified permits the common status page to be updated to correctly reflect the changed status. Subsequent modifications to the same page donot require amendment of the status page. The status page indicia are reset as part of the backup procedure and for ascertaining the page copy set for incremental copying. The ICBU LSN assumes one of two values as a function of the copy operation and another value for processing page modifications after the copy operation. A second number (ICRF.sub.-- LSN) is used in the restoration of a **file** after the **file** has been partially **restored** by a page merge in page number order from full and incremental copies. In this case, the ICRF.sub.-- LSN defines the point in the log for redo since the most recent copy was made.

6 Claims, 29 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 8

---------- KWIC ---------

Abstract Text - ABTX (1):

A method and means for achieving **files** of modifiable pages in a log based phased commit transaction management system (TMS) in which those pages which have been modified since the last full or **incremental backup** donot require during the copy operation any modifications to the page itself but merely to a common status page. This is accomplished by management of a pair of global log sequence numbers. Comparison between a first number (ICBU.sub.-- LSN) and each data page LSN as the page is modified permits the common status page to be updated to correctly reflect the changed status. Subsequent modifications to the same page donot require amendment of the status page. The status page indicia are reset as part of the backup procedure and for ascertaining the page copy set for incremental copying. The ICBU LSN assumes one of two values as a

function of the copy operation and another value for processing page modifications after the copy operation. A second number (ICRF.sub.-- LSN) is used in the restoration of a **file** after the **file** has been partially **restored** by a page merge in page number order from full and incremental copies. In this case, the ICRF.sub.-- LSN defines the point in the log for redo since the most recent copy was made.

Brief Summary Text - BSTX (2):
This invention relates to data preservation in an information handling system by full or incremental backup copying, where copying is non-disruptive of executing applications. More particularly, this invention relates to a method and means for archiving **files** of modifiable pages in a log based phased commit transaction management system (TMS).

Drawing Description Text - DRTX (4):
FIG. 3 shows a logical organization of a small **file** representing a database and a system catalog populated with assorted system constructs and data pages.

Drawing Description Text - DRTX (5):
FIG. 4 depicts the **file** after initially loading data and taking a required full copy thereof under the assumption that logging was suspended for the duration of initial loading of the database.

Drawing Description Text - DRTX (10):
FIGS. 9A-9B show the copies of **file** A on tape storage after full and incremental copy operations are archived.

Drawing Description Text - DRTX (11):
FIGS. 10A-10B depict **file** A after a recovery using the FC and IC of FIGS. 9A-9B and log records.

Drawing Description Text - DRTX (12):
FIGS. 11A-11B illustrate the state of the **file** and constructs after multiple updates to at least one page.

Detailed Description Text - DETX (11):
Referring now to FIG. 2, there is shown the time of occurrence relations among transaction primitives with reference to checkpoints and system failure. At the occurrence of failure and hence restart, the recovery manager must obtain the address of the most recent checkpoint record from a restart **file** or its equivalent, locate the checkpoint record in the system log, and proceed to search forward through the log from that point to the end. As a result of this process, the Recovery manager is able to determine both the transactions that need to be UNDOne (UNDO) and the transactions that need to be REDOne (REDO) in order to restore the resources to a consistent state.

Detailed Description Text - DETX (18):
For purposes of this invention, a data base includes data pages and ancillary constructs. In this regard, a **file** of pages is deemed to be a functional equivalent to a data base. The ancillary constructs include pages in the data base for tracking the allocation status etc. A page containing user data is called a "data page", a system-owned page tracking the allocation and

space availability status of data pages is called a "space map page" (SMP), and a system-owned page containing system related information is called a "header page".

Detailed Description Text - DETX (19):
As in DB2, a large table can be divided into many partitions each of which is a separate operating system **file**. Each partition would have a header page, one or more space-map pages and numerous data pages. Archive copies include copies of not only data pages but also those of SMPs and the header page.

Detailed Description Text - DETX (21):
In the header page, a field called "image.sub.-- copy.sub.-- bit.sub.-- update.sub.-- LSN" (ICBU.sub.-- LSN) is maintained. There is only one ICBU.sub.-- LSN for a **file**. The purpose of the ICBU.sub.-- LSN is that by using it, a transaction updating a data page can efficiently determine if the ICB in the SMP for that data page may need to be set. This is desirable because the ICB needs to be set only if it is the first update to a data page since the last copy operation. Accessing the SMP for subsequent updates of a data page must be avoided because of the overheads involved, such as, DASD I/O, locating and fixing the SMP in the buffer pool, latching it, searching for the appropriate ICB, and "unfixing it, just to determine that no change need be performed.

Detailed Description Text - DETX (23):
Associated with each archive copy is an copy roll forward LSN (ICRF.sub.-- LSN). During media recovery, this is the LSN from which the log would have to be scanned to identify log records whose updates might have to be REDOne to recover the data base after reloading the relevant archive copies (the latest full copy and any subsequent incremental copies). ICRF.sub.-- LSN is remembered in a system catalog along with the information such as the name of the **file** which contains the archive copy, and whether a full or incremental copy was taken.

Detailed Description Text - DETX (36):
A system catalog is a directory of **files** with reference to their location. It also may contain status information. As used in this invention, it includes the **file** name, whether it is a full or incremental copy, date and time, device where stored, and the associated ICRF.sub.-- LSN.

Detailed Description Text - DETX (57):
Copying the ith data page into the archive copy **file**.

Detailed Description Text - DETX (100):
Referring now to FIG. 3, there is shown a logical **file** and page organization as used in the invention. More particularly, a **file** or data base includes a header page, a space map page (SMP), and a plurality of data pages (d-pages 1-4). The data, arbitrarily named "**file** A", is loaded into the database 13 in the DASD external storage under control of a CPU controlled load operation. To save log space and logging overhead, such loading is performed with the logging suspended. Also, no concurrent updates are permitted during execution of the loading operation. As part of the operation, page numbers are sequentially assigned to each page. The header page is number page 0, the SMP is page 1, while d-page 1-4 are numbered as pages 2-5. Now, the page numbers merely define a local page sequence and should NOT be confused with the page.sub.--

LSN's. The latter are pointers embedded in each page when written through to DASD storage after being logged which denote the position of the REDO/UNDO records in the log containing the most recent page modification. Initially, the page.sub.-- LSN's are set to 0. Likewise, the ICB's in the SMP for all the d-pages are set to 0 as is the ICBU.sub.-- LSN in the header page. Lastly, since no backup exists for **file** A, there is no entry in the system catalog.

Detailed Description Text - DETX (101):
As seen in FIG. 3, the system catalog broadly contains a location pointer and other archive or backup copy information for an either a full or incremental copy of **file** A and other **files**. Typically, such archive copies would be stored on an auxiliary store such as an automated tape library. Any recovery use of the backup copy would require access of the tape library and staging therefrom to DASD storage.

Detailed Description Text - DETX (102):
Referring now to FIG. 4, there is shown the state of **file** A and the system catalog after a Full copy (FC) has been taken of **file** A subsequent to initial loading and prior to any modifications being permitted to the data pages. Note, in FIG. 4, the ICBU.sub.-- LSN and the ICRF.sub.-- LSN have been set to 100 since that was the current end.sub.-- of.sub.-- log after the completion of the FC. The header page is subsequently updated by the method to record this value of the ICBU.sub.-- LSN by writing a log record having the LSN of 110. That is, the location in the log where the header page stores the current ICBU.sub.-- LSN value of 100 is log location 110.

Detailed Description Text - DETX (103):
Referring now to FIG. 5, there is shown the system catalog entry wherein the FC of **file** A was taken at 23:50 on Jan. 5, 1992 and stored in the tape library address T001. In case of a loss of DASD D001 (not shown) after the FC has been taken, then the DBMS recovery operation for **file** A would include the steps of:

Detailed Description Text - DETX (104):
(a) ascertaining the current end of log (arbitrarily termed END.sub.-- LSN) and the tape location for the most recent full copy of **file** A from the system catalog;

Detailed Description Text - DETX (105):
(b) loading the FC backup of **file** A from tape location T001 to a DASD D002 (not shown);

Detailed Description Text - DETX (106):
(c) ascertaining any incremental copies (IICs) to **file** A subsequent to the FC from the system catalog and update accordingly (In the instant case for step (c), no IICs were as yet taken subsequent to the FC for **file** A);

Detailed Description Text - DETX (107):
(d) positioning the system log at the ICRF.sub.-- LSN=100 since that is the most recent backup and applying log records for updates of the pages of **file** A on DASD0002 if the page.sub.-- LSN<LSN of the log record (In the posited instance, this would be from the ICRF.sub.-- LSN=100 to END.sub.-- LSN).

Detailed Description Text - DETX (108):
In order to reduce the number of log scans during step (d), the DBMS could utilize another directory located in a modified systems catalog. This other directory preferably would consist of log ranges during which **file** A was opened for update. This directory labeled "SYSLGRNG" (not shown) for the instant example would have an entry for **file** A indicative of LSNs from 100 to 110 for the header page update. The recovery operation would then be complete.

Detailed Description Text - DETX (110):
In the update of d-page 1, the ICBU.sub.-- LSN=100 for **file** A is also recorded in the data base control block (DBCB) per FIG. 6A. Since the page.sub.-- LSN of d-page 1 of 0<ICBU.sub.-- LSN of 100, then the update method has to set the ICB1 in the SMP prior to updating d-page 1. Both the ICB1 change and the LSN where the change is recorded in the log (LSN212) are marked on the SMP as shown in FIG. 6B. Next, the value of d-page 1 is changed from A1 to B1 and log recorded at LSN 215. These changes are indicated in the updated d-page 1 in FIG. 6C.

Detailed Description Text - DETX (112):
Referring now to FIG. 7, there is set forth **file** A as updated and written to DASD including SMP and page update changes.

Detailed Description Text - DETX (113):
Referring now to FIGS. 8A-8K, there is set forth the IC for **file** A depicted in FIG. 7. FIGS. 8A and 8B signify the initial state of **file** A for the IC operation per the DBCB and the SMP. The first step is to read the SMP into the buffer pool. Next, the ICBU.sub.-- LSN is changed to a maximum value "FFFF..FF" in hexidecimal notation as per FIG. 8C. Following this the ICRF.sub.-- LSN is set to the current end.sub.-- of.sub.-- log=463. The system catalog is now updated as expressed in FIG. 8D with a new entry indicative that an IC is being taken, date-time, location of the IC in the tape, and the ICRF.sub.-- LSN.

Detailed Description Text - DETX (117):
Significantly, if the ICBU.sub.-- LSN value had remained at 100, then the update to d-page 1 would NOT have resulted in ICB1 being set to "1". Thus, subsequent IICs would not copy d-page 1 with the value "CI". Any completion of a subsequent IC would result in an ICRF.sub.-- LSN>504. In the case of any loss of **file** A after the subsequent IC, the restoration of backup copies using an ICRF.sub.-- LSN >504 would cause the update of LSN 504 to be missed. This explains the need for setting the ICBU.sub.-- LSN to a maximum value DURING THE IC while a new non-maximum value is being established. Setting ICBU.sub.-- LSN to a non-max value AFTER completing the IC including latching, copying, and unlatching all pages in the copy set is required, otherwise every update to any page in the **file** would cause access to the SMP for setting the corresponding ICB or checking whether the ICB is already set. The latter is considered wasteful.

Detailed Description Text - DETX (124):
Referring now to FIGS. 9A-9B, there is shown the of **file** A after full and incremental copy operations as archived on tape storage. Suppose the DASDs storing the current copies failed such that it was necessary to recover the current state of **file** A. In this case, according to the system catalog as shown in FIG. 8D, the location of the most recent full copy and subsequent incremental copies is established and the ICRF.sub.-- LSN=463. This means

restoring **file** A based on merging the full and incremental copies by PAGE NUMBERS presently archived respectively at tape library addresses T001 and T002.

Detailed Description Text - DETX (125):
Referring now to FIG. 10A, there is shown the page merged copy of **file** A from the stored tapes. To bring the state of the page forward, the log would be scanned from LSN 463 and would apply the updates to the SMP at LSN 490, d-page 1 at LSN 504, header page at 530. This yields the final **file** state as shown in FIG. 10B.

Detailed Description Text - DETX (126):
Referring now to FIGS. 11A-11B, there is shown the state of the **file** and constructs after multiple updates to at least one page. In this case, a transaction updates d-page 4 and then d-page 1.

Claims Text - CLTX (1):
1. A method for archiving a **file** of modifiable pages in a phased commit transaction management system (TMS) having a log, a processor, and a storage subsystem in which pages stored therein are staged to and from the processor, said **file** having a header page, status page (SMP), and at least one data page, the processor being responsive to each transaction and executing selective modifications to predetermined ones of the pages, each modification to a page state being recorded in the log and assigned an ascending log sequence number (page.sub.-- LSN), said page.sub.-- LSN being recorded on the page, comprising the steps of:

Claims Text - CLTX (8):
2. The method according to claim 1, wherein each page in the **file** has recorded thereon an ascending page number independent of any other indicia, wherein said system catalog comprises entries recording information about each full or incremental copy of the **file**, each entry including a pointer to the location of each archive copy of the **file** in the storage subsytem and an ICRF.sub.-- LSN set to the end.sub.-- of.sub.-- log.sub.-- LSN as per claim 1, step (e), and wherein said method further comprises the steps of:

Claims Text - CLTX (9):
(g) making a full copy of each **file** each time the data is loaded into the **file** without logging, suspending any data page modifications until the full copy is made, and recording the pointer location, copy type (full) and ICRF.sub.-- LSN in the system catalog;

Claims Text - CLTX (10):
(h) making incremental or full copies of each **file** on a scheduled or opportunistic basis including entries to the system catalog;

Claims Text - CLTX (11):
(i) responsive to the unavailability of a **file** in the external storage subsystem, recovering said **file** by merging according to their page numbers the contents of the full and incremental copies as defined by the system catalog, and modifying the merged page set according to the log records having an LSN equal to or greater than the ICRF.sub.-- LSN.

Claims Text - CLTX (30):
6. A method for archiving a **file** of modifiable pages in a phased commit transaction management system (TMS) having a log, a processor, and a storage subsystem in which pages stored therein are staged to and from the processor, said **file** having a header page, status page (SMP), and at least one data page, each page in the **file** has recorded thereon an ascending page number independent of any other indicia, the processor being responsive to each transaction and executing selective modifications to predetermined ones of the pages, each modification to a page state being recorded in the log and assigned an ascending log sequence number (page.sub.-- LSN), said page LSN being recorded on the page, comprising the steps of:

Claims Text - CLTX (32):
(b) setting the ICBU.sub.-- LSN to a maximum value, and establishing a second global log sequence number (ICRF.sub.-- LSN) and recording said ICRF.sub.-- LSN in a system catalog, said system catalog comprising entries recording information about each full or incremental copy of the **file**, each entry including a pointer to the location of each archive copy of the **file** in the storage subsystem and an ICRF.sub.-- LSN set to an end.sub.-- of.sub.-- log.sub.-- LSN,

Claims Text - CLTX (37):
(g) making a full copy of each **file** each time the data is loaded into the **file** without logging, suspending any data page modifications until the full copy is made, and recording the pointer location, copy type (full) and ICRF.sub.-- LSN in the system catalog;

Claims Text - CLTX (38):
(h) making incremental or full copies of each **file** on a scheduled or opportunistic basis including entries to the system catalog; and

Claims Text - CLTX (39):
(i) responsive to the unavailability of a **file** in the external storage subsystem, recovering said **file** by merging according to their page numbers the contents of the full and incremental copies as defined by the system catalog, and modifying the merged page set according to the log records having an LSN equal to or greater than the ICRF.sub.-- LSN.